NuMI and Rapid Transfer BPM
Test Application Program Specifications

Introduction

The NuMI and Rapid Transfer BPM systems are derived from hardware and software elements used in the Recycler Ring BPM system. In the Recycler BPM system the concept of performing a system calibration by making statistical calibration measurements to determine gain, offset and timing correction factors seemed to some to be possible. In the NuMI and Rapid Transfer systems the signal generator and other analog hardware used to check instrumentation integrity do not provide sufficient accuracy to support such calibration. In these BPM systems we use the term test rather than calibration. So we require a 'test application' that exercises the self-test features of the BPM front-end and presents to the user various graphical displays of measured test data.

The test application must support the NuMI and Rapid Transfer beam line BPM systems. (It is possible that other systems will need to be supported in the future.) The NuMI BPM system is the least complicated test system and the first requiring support.

The test application must present a user interface that is easy to understand and use. The test application must by default present a 'User' mode with a minimal set of options for use in daily operations. The test application must also provide an 'Expert' mode that allows more options and a greater degree of control for the Engineering staff. The goal is to provide as close to a 'one button' measurement as possible for normal User mode operation.

The initial version of the test application will be run interactively by the user. The design should take into account the requirement of an automated periodic test capability. This feature will be added after the manual test mode is fully operational.

In the discussion that follows the use of the term 'calibration' is necessary to correctly reference various essential data structures and concepts that were established in the Recycler BPM system from which the NuMI and Rapid Transfer systems were derived. The terms calibration and test can be considered synonymous in this context.

Program Operation

Refer to the straw man display below for a suggested page layout.

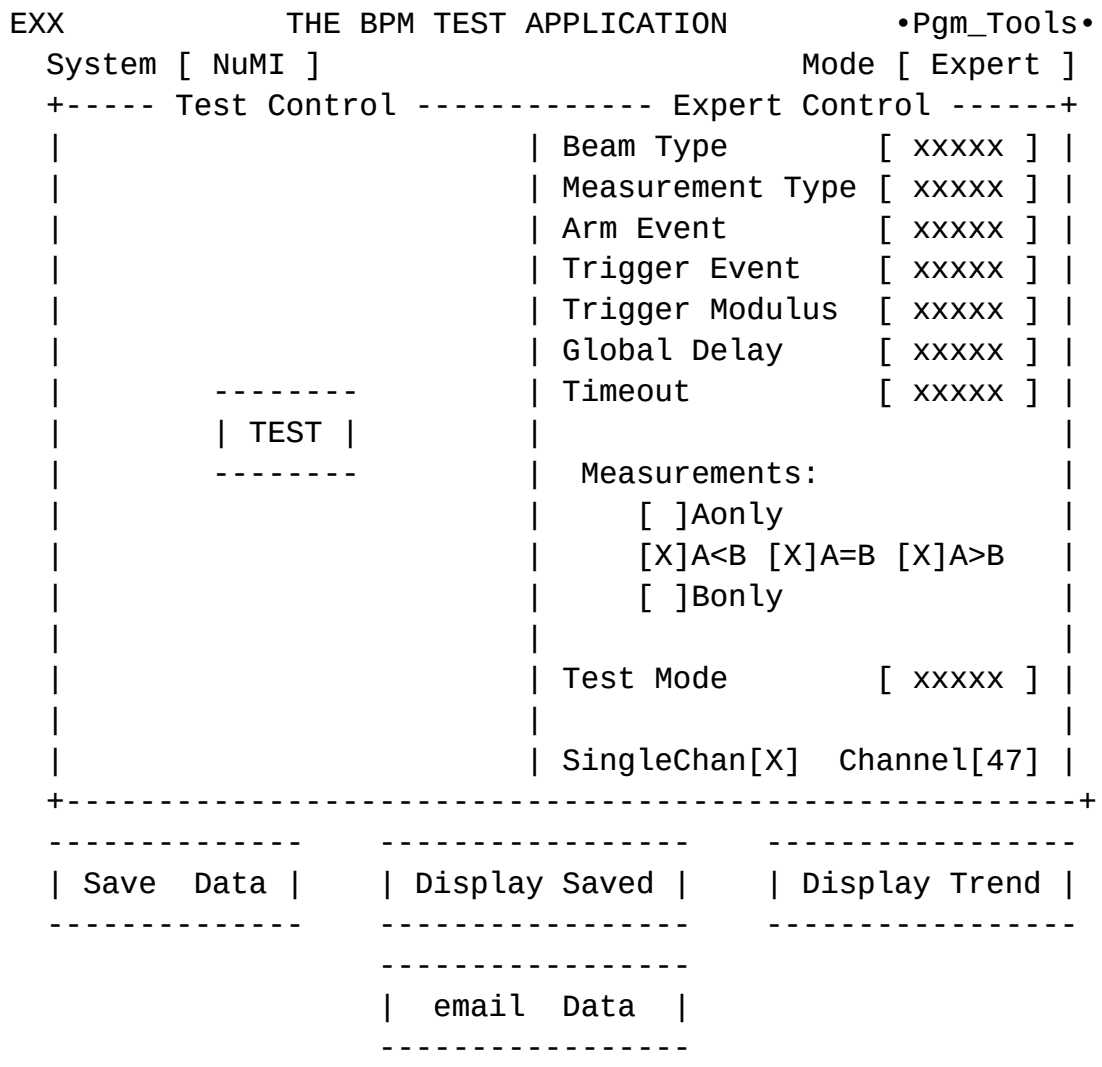

Figure 1 - Straw man Display Example

When entered, and before supporting any other user interaction, the test application must present a pop-up menu allowing the choice of BPM system. The default menu selection could be the last system tested at the current console, but this is not required. After the user selects the desired BPM system the test application activates the main screen in User mode. The main screen contains one or more of the following items depending upon the states of the User/Expert mode and the test measurement:

•A System selection button. This button presents the same system selection pop-up menu that was presented when the test application was entered. When activated this button must assure that any active test measurement is aborted in a clean manner and then switch to the main screen for the newly selected BPM system in User mode.

• A User/Expert mode button. The User/Expert button toggles between the two modes. Selection of Expert mode should be challenged (e.g., Are you sure?) and confirmed by the user. Selection of User mode does not require such a challenge. When activated this button disables/enables the Expert mode and Expert control region described below. When transitioning from one mode to the other the test application must abort any active test measurement in a clean manner before proceeding.

• A Test Control region that contains all User mode controls relating to the test measurement. The test control region must present the minimal set of test options required by the selected BPM system, and a "TEST" button. The TEST button must cause the test measurement to be invoked with the test options selected by the user and default values for all other test parameters. See the Test Measurement section below for a detailed description of the test procedure. The test application must present an "ABORT" button whenever the test measurement is active. When selected the ABORT button should abort the active test measurement in a clean manner and then 'go away'. The test application must provide a display that indicates the state and status of the test measurement as it progresses. The test application must display the acquired test data as they are collected using the Primary Data Display described below.

• An Expert control region that contains additional controls which allow the expert user to select desired values for test parameters that contain default values when operating in the User mode. The expert control region must be presented only in response to activation of the Expert mode with the User/Expert mode button described above.

• A Save Data button that allows the user to save a test data set for future display and evaluation. This button must be presented only when in User mode and a complete set of fresh test data is available. (Data collected in Expert mode should not be saved since it may not be in the standard form.) When the Save Data button is activated the test application must solicit a file comment from the operator to be used to easily identify the data set. The following items should be saved with each data set:

- date and time from data time stamp,
- data type tag,
- file comment,
- all data header information,
- all position and intensity quantities for Value, Min, Max and RMS.

Note that since NuMI and Rapid Transfer require different amounts of data for each test the structure of the saved data will change from system to system. (Hence the data type tag.)

• A Display Saved Data button that allows the user to view previously saved data on the Primary Data Display described below. This button must be presented only when a test measurement is not in progress. This button must check to see if there is valid unsaved data and offer the user the opportunity to save that data before proceeding. The button must present a pop-up menu of all previously saved data sets. Menu items must contain the file comment and the date and time from the data time stamp.

• A Display Trend Data button that allows the user to view data that was saved over time on the Trend Data Display described below. This button must be presented only when a test measurement is not in progress. This button must check to see if there is valid unsaved data and offer the user the opportunity to save that data before proceeding. The button must present a pop-up menu allowing the selection of one of two (possibly more) trend modes:

- single location versus time and
- multiple location versus time.

If the single location mode is selected the test application must present pop-up menus soliciting selection of the BPM location, a start date/time and an end date/time. If the multiple location mode is selected the test application must present pop-up menus soliciting selection of the three BPM locations as well as a start date/time and an end date/time.

• An Email Data button that allows the user to send the currently displayed test data to a specified email account. This button must be presented only when there is a valid data set visible. The email content should include all of the information that would be saved if the Save Data button were pressed. The email data should be comma separated for use with spreadsheet applications.

Primary Data Display

The primary display is a stacked graph with position on the top and intensity below. Each graph must contain data for all channels in the system at each of the three test settings with channel location along the X-axis. Each of the graph's position/intensity points must be plotted with the Min/Max and RMS values shown as error bars.

Trend Data Display

The trend display for saved test data is a stacked graph with position on the top and intensity below. There are two data views each as a function of time. The first view is of

a specified BPM location at each of the three test settings. The second view is of three specified BPM locations at a specified test setting. Each of the graph's position/intensity points must be plotted with the Min/Max and RMS values shown as error bars.

Test Measurement

The BPM front-end supports a 'Test' mode that takes sixteen flash measurements at a specified periodic rate and stores the data in a turn-by-turn data structure. Since the data are stored as a turn-by-turn measurement it is possible for the front-end to provide minimum, mean, maximum and RMS values for both position and intensity. The Test measurement is a one-shot triggered measurement that happens to take multiple samples. In all systems the second measurement event (1 of 0..15) is reserved for testing.

To take a test measurement the test application must:

1 - send an Acquisition Specification to invoke the measurement,
2 - send a Test Specification to establish test conditions,
3 – wait for the measurement to complete,
4 – read the data from the Test data device and
5 – loop to step #2 with new test conditions.

The complete test procedure cycles through three test conditions (A<B, A=B and A>B) displaying the data as they are collected. When all three measurements have been completed the Acquisition Specification enabled in step #1 must be disabled. It is not necessary to restore the Acquisition or Calibration specifications to their original values. Also, it is not necessary to send the setting property of the Acquisition Specification to disable the test measurement. Simply setting the basic control property of the Acquisition Specification to off will disable the test measurement.

The Calibration and Acquisition Specifications differ from system to system. The details of the specification structures are presented in the system requirement sections below. The method used to wait for the measurement to complete will depend upon the arm and trigger conditions specified in the Acquisition Specification. Typically the data are read two to three 15 Hz. ticks after a Tclk arm event. The Test data are read using the data readout algorithm described in "Event Driven Data Acquisition for the Recycler Ring BPM Front-end" which is part of Beams-doc-1169.

The ACNet devices required to complete a test measurement include:

Acquisition Specification – second element (1 of 0..15) of device m:BPxACQ,
Test Specification – m:BPxTST,

Readout Specification – m:BPxTMS and
Test Data – m:BPxTMV
where m indicates the machine and x the front-end node.

The Readout Specification must contain the following field values.

```
_eventIndex              1
_dataType                0
_beginTurn               1
_numTurns                16
_channel                 0
```

Expert Mode

The Expert mode enables various controls that provide additional control over the test procedure. The following Acquisition Specification fields should be made available to the expert user:

```
_beamType,
_measurementType,
_armEvent,
_triggerEvent,
_triggerModulus
_globalDelay  and
_timeout.
```

A set of controls must be included to allow the user to specify which test measurement or measurements (e.g., A<B, A=B or A>B) are to be taken when the next test is performed.

In the Expert mode the user may want to concentrate on a single channel of the system. A mechanism must be provided for determining which channel is desired. In this case the test application should collect and display, in a second graph, the turn-by-turn data for the specified channel. The turn-by-turn data will provide position and intensity values for each of the sixteen data samples used to produce the normal Test values. Turn-by-turn data should be plotted as a stacked graph with position on the top and intensity on the bottom. Each graph should contain the data value versus sample number for the sixteen samples.

The ACNet devices required to read the turn-by-turn data include:
Readout Specification – m:BPxTBS and
Turn-by-turn Data – m:BPxTBN
where m indicates the machine and x the front-end node.

The Expert mode must provide a mechanism for deleting specified records from the Saved Data archive.

NuMI Requirements

The NuMI test system's Calibration Specification has the following definition.

```
typedef enum {      // identifies system hardware configuration
  kCalRecyclerStyle = 0,
  kCalNuMIStyle,
  kCalRapidTransferStyle
} eCalibrationSystemType;

typedef enum {      // CalibrationSpecification's _frequency member
  kCalibrationFrequencyMin = 0,
  kCalibration53 = kCalibrationFrequencyMin,
  kCalibration2Point5,
  kNumCalibrationFrequencys,
  kCalibrationFrequencyMax = kNumCalibrationFrequencys - 1,
  kCalibrationFrequencyDefault = kCalibration53
} eCalibrationFrequency;

typedef enum {      // CalibrationSpecification's _control member
  kCalibrationControlMin = 0,
  kCalibrationDisabled = kCalibrationControlMin,
  kAOnly,
  kAGreaterThanB,
  kAEqualB,
  kALessThanB,
  kBonly,
  kNumCalibrationControls,
  kCalibrationControlMax = kNumCalibrationControls - 1,
  kCalibrationControlDefault = kCalibrationDisabled
} eCalibrationControl;

class CalibrationSpecification { // specify test conditions
    eCalibrationSystemType _calibrationSystemType : 4;
    unsigned long int      _mustBeZero : 12;
    eCalibrationFrequency  _frequency : 8;
    eCalibrationControl    _control : 8;
} __attribute ((packed));
```

The frequency field should always be `kCalibration53`. Normal User mode tests use only the A<B, A=B and A>B control field values, the A-only and B-only values should be made available as options in Expert mode measurements. Note that since the frequency field is not changed there is nothing for the test application's user to specify in User mode other than the request to make a test measurement.

The NuMI test system's Acquisition Specification has the following values.

Event Index 1 - Test:

_enable	**kAcquisitionOn**
_measurement	**kTestMeasurement**
_beamMode	**kProton**
_beamType	**kInjectExtract**
_measurementType	**k6Batch53MhzTimed (0)**
_armEvent	**0x26**
_triggerEvent	**0xAA**
_pretriggerEnable	**kPretriggerDelayOff**
_triggerModulus	**50**
_globalDelay	**0**
_intensityThreshold	**0.0**
_timeout	**1**

This test measurement is configured to acquire its sixteen samples upon receipt of the Tclk 26 arm event (which represents "End of MI Ramp Flattop" when there is no longer any beam in the machine) plus 150 mSec. The trigger modulus value of 50 causes the sixteen measurements to be taken at ~2000 Hertz (every 50 turns) and the entire measurement will be completed within ~200 mS. The test data should be available for readout within five 15 Hz. ticks after the Tclk 26 arm event.

Rapid Transfer Requirements

The Rapid Transfer system's Calibration Specification has the following definition.

```
typedef enum {      // identifies system hardware configuration
  kCalRecyclerStyle = 0,
  kCalNuMIStyle,
  kCalRapidTransferStyle
} eCalibrationSystemType;

typedef enum {      // CalibrationSpecification's _mode member
  kCalibrationModeMin = 0,
  kCalibrationContinuous = kCalibrationModeMin,
  kCalibrationPulse,
  kNumCalibrationModes,
  kCalibrationModeMax = kNumCalibrationModes - 1,
  kCalibrationModeDefault = kCalibrationPulse
} eCalibrationMode;

typedef enum {      // CalibrationSpecification's _control member
  kCalibrationControlMin = 0,
  kCalibrationDisabled = kCalibrationControlMin,
  kAGreaterThanB,
  kAEqualB,
```

```
  kALessThanB,
  kNumCalibrationControls,
  kCalibrationControlMax = kNumCalibrationControls - 1,
  kCalibrationControlDefault = kCalibrationDisabled
} eCalibrationControl;

class CalibrationSpecification { // specify test conditions
    eCalibrationSystemType _calibrationSystemType : 4;
    unsigned long int      _mustBeZero : 12;
    eCalibrationMode       _mode : 8;
    eCalibrationControl    _control : 8;
} __attribute ((packed));
```

The mode field should always be `kCalibrationPulse` under normal test conditions but the other values for the mode field should be made available in Expert mode measurements.

The Rapid Transfer system requires two test measurements, one at 53 MHz and another at 2.5 MHz. The two Acquisition Specifications have the following values.

Event Index 1 – 53 MHz Test:

_enable	**kAcquisitionOn**
_measurement	**kTestMeasurement**
_beamMode	**kCalibration**
_beamType	**k53MHz**
_measurementType	**k53MhzBatch84 (4 of 0..8)**
_armEvent	**0x26**
_triggerEvent	**0xAA**
_pretriggerEnable	**kPretriggerDelayOff**
_triggerModulus	**100**
_globalDelay	**0**
_intensityThreshold	**0.0**
_timeout	**1**

Event Index 1 – 2.5 MHz Test:

_enable	**kAcquisitionOn**
_measurement	**kTestMeasurement**
_beamMode	**kCalibration**
_beamType	**k2point5MHz**
_measurementType	**k2.5MhzEnsemble (1 of 0..8)**
_armEvent	**0x26**
_triggerEvent	**0xAA**
_pretriggerEnable	**kPretriggerDelayOff**
_ triggerModulus	**100**
_globalDelay	**0**
_intensityThreshold	**0.0**
_timeout	**1**

These test measurements are configured to acquire the sixteen samples upon receipt of the Tclk 26 arm event. The trigger modulus value of 50 causes the sixteen measurements to be taken at ~2000 Hertz (every 50 turns) and the entire measurement will be completed within ~130 mS. The test data should be available for readout within five 15 Hz. ticks after the Tclk 26 arm event.

End.